Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 09.10.2023 10:10:21
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

Факультет информационных технологий

УТВЕРЖДЕНО
Декан факультета
Информационных технологий
_____ / Д.Г. Демидов /
«16» ____02_____ 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Методология и практики DevOps»

Направление подготовки
09.03.01 Информатика и вычислительная техника

Профиль
«Кибербезопасность автоматизированных систем»

Квалификация
Бакалавр

Формы обучения
очная

Москва, 2023 г.

Разработчик(и):

Должность, звание / _____ /

Согласовано:
И.о. заведующего кафедрой «Информационная безопасность»,  /А.Ю. Гневшев/

Руководитель образовательной программы,  /А.Ю. Гневшев/

Содержание

1 Цели, задачи и планируемые результаты обучения по дисциплине 4
2 Место дисциплины в структуре образовательной программы 7
3 Структура и содержание дисциплины 7
3.1 Виды учебной работы и трудоемкость 7
3.2 Тематический план изучения дисциплины 8
3.3 Содержание дисциплины 8
4 Учебно-методическое и информационное обеспечение 8
4.1 Основная литература 8
4.2 Дополнительная литература 8
5 Материально-техническое обеспечение 8
5.1 Требования к оборудованию и помещению для занятий 8
5.2 Требования к программному обеспечению 9
6 Методические рекомендации 9
6.1 Методические рекомендации для преподавателя по организации обучения 9
6.2 Методические указания для обучающихся по освоению дисциплины 9
7 Фонд оценочных средств 9
7.1 Методы контроля и оценивания результатов обучения 9
7.2 Шкала и критерии оценивания результатов обучения 10
7.3 Оценочные средства 18
7.3.1 Экзаменационные вопросы 18

1 Цели, задачи и планируемые результаты обучения по дисциплине

К основным целям освоения дисциплины следует отнести:

- формирование у слушателей знаний и навыков по методологии DevOps для активного взаимодействия специалистов по разработке со специалистами по информационно-технологическому обслуживанию и взаимной интеграции их рабочих процессов для обеспечения качества продукта.

К основным задачам освоения дисциплины относится освоение следующих тем:

- разбирается жизненный цикл (ЖЦ) программного обеспечения;
- роль DevOps-инженера в ЖЦ;
- программные инструменты DevOps.

Обучение по дисциплине «Методология и практики DevOps» направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенций	Индикаторы достижения компетенции
ОПК-7. Способен участвовать в настройке и наладке программно аппаратных комплексов	ИОПК-7.1. Знает основные языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения. ИОПК-7.2. Умеет составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули. ИОПК-7.3. Владеет языком программирования, методами отладки и тестирования работоспособности программы
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	● ИОПК-8.1. Знает основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной системы. ● ИОПК-8.2. Умеет проектировать блок-схемы алгоритмов, оценивать производительность алгоритмов и затраты памяти на работу алгоритма, разрабатывать программы на основе спроектированного алгоритма и проводить отладку программы, применять методы системного анализа и математического моделирования при разработке и эксплуатации ИС, проводить структурный анализ, функциональный анализ, объектноориентированный анализ иерархии классов. осуществлять

	организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла информационной системы. ● ИОПК-8.3. Владеет навыками разработки программ, построения блок-схем алгоритмов и оценки производительности алгоритмов, работы с унифицированным языком визуального моделирования, составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла.
ПК-5. Способен разрабатывать компоненты системных программных продуктов	● ИПК-5.1. Знает: архитектуру аппаратной платформы, синтаксис, особенности программирования и стандартные библиотеки языков программирования Python, C/C++, Java, Golang ; Стандарты реализации интерфейсов подключаемых устройств Технологии разработки драйверов Системы прерываний и адресации памяти ОС Технологии разработки и отладки системных продуктов. Комплекты средств разработки целевой ОС Система команд микропроцессора целевой аппаратной платформы Принципы кросс - платформенного программирования Конструкции распределенного и параллельного программирования Принципы организации, состав и схемы работы ОС Принципы управления ресурсами Методы организации файловых систем Принципы построения сетевого взаимодействия Архитектура и принципы функционирования коммуникационного оборудования Стандарты информационного взаимодействия систем Методики тестирования разрабатываемого ПО Локальные нормативные правовые акты, действующие в организации Государственные стандарты Единой системы программной документации (ЕСПД) Способы адресации памяти целевой

	аппаратной платформы Технологии разработки компиляторов Методы и основные этапы трансляции Технологии программирования и разработки блок -схем Основы применения теории алгоритмов Принципы организации, состав и схемы работы операционных систем Основы архитектуры, устройства и функционирования вычислительных систем Средства программирования и их классификация Архитектура сред программирования Классификация языков программирования Основные структуры данных. Основные модели данных и их организация Принципы объектно - ориентированного программирования Языки функционального и логического программирования Конкурентное программирование Методы и алгоритмы грамматического разбора текста программы Компиляторы языков программирования, их виды, принципы работы Методы и алгоритмы генерации исполняемого кода Структура объектных и исполняемых файлов в целевой операционной системе Компоновщики, методы сборки исполняемых файлов из объектных файлов Методы и алгоритмы оптимизации исполняемого кода Интерпретаторы языков программирования, их виды, принципы работы Методы управления памятью Принципы работы программного отладчика ● ИПК -5.2. Умеет: Применять языки программирования, определенные в техническом задании на разработку драйвера, для написания программного кода Создавать блок -схемы алгоритмов функционирования разрабатываемых программных продуктов Оценивать вычислительную сложность

	алгоритмов функционирования разрабатываемых программных продуктов Работать со стандартными контроллерами устройств (графическим адаптером, клавиатурой, мышью, сетевым адаптером) Работать с документацией, прилагаемой разработчиком устройства Осуществлять отладку драйверов устройств для операционной системы Применять языки целевой аппаратной платформы, определенной в техническом задании на разработку, для написания программного кода Применять технологию разработки компиляторов Осуществлять отладку программных продуктов для целевой операционной системы Осуществлять отладку утилит операционной системы Разрабатывать программный код на языках программирования низкого уровня Осуществлять отладку программ, написанных, загрузчиков, сборщиков Разработкой эксплуатационной документации на разработанных драйверов, компиляторов, загрузчиков, сборщиков Сопровождением разработанных драйвера устройства, компиляторов, загрузчиков, сборщиков Получением технической документации по языку программирования, системе команд процессора устройства, адресации памяти и регистров процессора устройства Изучением технической документации по языку программирования, системе команд процессора устройства, адресации памяти и регистров процессора устройства Разработка блок -схемы компиляторов, загрузчиков, сборщиков Написание исходного кода компиляторов, загрузчиков, сборщиков Отладка компиляторов, загрузчиков, сборщиков Реинжиниринг разработанных компиляторов,

	загрузчиков, сборщиков, драйвера устройства.
ПК-6. Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности	● ИПК-6.1. Знает: теоретические основы проектирования крупного масштаба и сложных систем; стандарты оформления технических заданий; методы концептуального, функционального и логического проектирования систем; методы тестирования; международные стандарты на структуру документов. ● требований; нормативные и методические материалы по созданию документов требований к системам ; методы оценки качества программных систем; способы масштабирования информационных систем для учета их при логическом проектировании. ● ИПК-6.2. Умеет: формулировать цели, исходя из анализа проблем, потребностей и возможностей; разрабатывать технико - экономическое обоснование; декомпозировать функции на подфункции; алгоритмизировать деятельность; разрабатывать структуры типовых документов; исполнять ручные тесты, проектировать и разрабатывать сложные системы; использовать основные приемы web-дизайна. Внедрять графические, звуковые, анимационные объекты в систему; формировать интерактивные блоки web-ресурса; разрабатывать модели концептуальной, функциональной и логической архитектуры системы; спроектировать информационную систему для заданного предприятия по заданным характеристикам с помощью конфигурирования и программирования. ● ИПК -6.3. Владеет: навыками концептуального, функционального и логического проектирования; средствами автоматизации проектирования ПО, работы со средствами Internet и Web - технологий для решения задач

	профессиональной деятельности; навыками проектирования схемы последовательностей, состояний и взаимодействий компонентов системы.

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

2 Место дисциплины в структуре образовательной программы

Дисциплина относится к базовой части основной профессиональной образовательной программы бакалавриата.

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ОПОП:

- Автоматизация процессов жизненного цикла программных средств.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы, т.е. 144 академических часа (из них 72 часа – самостоятельная работа студентов и 72 часа – аудиторные занятия).

Разделы дисциплины изучаются в пятом семестре обучения, т.е. на третьем курсе. Форма контроля – экзамен.

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры	
			Семестр	Неделя семестра
1	**Аудиторные занятия**	**72**	5	1-18
	В том числе:			
1.1	Лекции	4	5	1-18
1.2	Семинарские/практические занятия			
1.3	Лабораторные занятия	68	5	
2	**Самостоятельная работа**	**72**	5	1-18
3	**Промежуточная аттестация**		5	19-121
	Экзамен			
	Итого:	**144**		

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/ практические занятия	Лабораторные занятия	Практическая подготовка	
1	Тема 1. DevOps	24	1		12		12
2	Тема 2. Системы виртуализации и контейнеризации	24			10		12
3	Тема 3. Системы непрерывной интеграции	24	1		12		12
4	Тема 4. Системы управления конфигурацией. Инфраструктура как код	24	1		12		12
5	Тема 5. Системы оркестрации	24	1		12		12
6	Тема 6. Системы мониторинга.	24			10		12
Итого		**144**	**4**		**68**		**72**

3.3 Содержание дисциплины

1. DevOps
2. Системы виртуализации и контейнеризации
3. Системы непрерывной интеграции
4. Системы управления конфигурацией. Инфраструктура как код
5. Системы оркестрации
6. Системы мониторинга.

4 Учебно-методическое и информационное обеспечение

4.1 Основная литература

1. Философия DevOps – Дженнифер Дэвис, Кэтрин Дэниелс – O’Reilly.

4.2 Дополнительная литература

1. Руководство по DevOps. Как добиться гибкости, надежности и безопасности мирового уровня в технологических компаниях - Джез Хамбл, Джон Уиллис, Патрик Дебуа.

2. Ускоряйся! Наука DevOps. Как создавать и масштабировать высокопроизводительные цифровые организации - Джез Хамбл, Джин Ким, Николь Форсгрен.

5 Материально-техническое обеспечение

5.1 Требования к оборудованию и помещению для занятий

Практические занятия (семинары) и самостоятельная работа студентов должна проводиться в специализированных аудиториях с комплектом мультимедийного оборудования и/или доской для записей материалов. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов.

5.2 Требования к программному обеспечению

Для проведения практических занятий (семинаров) специального программного обеспечения для освоения дисциплины не требуется.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

6.2 Методические указания для обучающихся по освоению дисциплины

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются аудиторные занятия, семинары и практики.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;

- контроль со стороны преподавателей (текущий и промежуточный).

Текущий контроль осуществляется на аудиторных занятиях, промежуточный контроль осуществляется на зачете в письменной (устной) форме.

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность компетенций;
- оформление материала в соответствии с требованиями.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- экзамен.

7.2 Шкала и критерии оценивания результатов обучения

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине.

Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
ПК-5. Способен разрабатывать компоненты системных программных продуктов				
ИПК-5.1. Знает: архитектуру аппаратной платформы, синтаксис, особенности программирования и стандартные библиотеки языков программирования Python, C/C++, Java, Golang ; Стандарты реализации интерфейсов подключаемых устройств; технологии разработки драйверов; системы прерываний и	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3).	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины «Знать» (см. п. 3). Свободно оперирует приобретенными знаниями.

адресации памяти ОС; технологии разработки и отладки системных продуктов; комплекты средств разработки целевой ОС; система команд микропроцессора целевой аппаратной платформы; принципы кросс-платформенного программирования; конструкции распределенного и параллельного программирования; принципы организации, состав и схемы работы ОС; принципы управления ресурсами; методы организации файловых систем; принципы построения сетевого взаимодействия; архитектура и принципы функционирования коммуникационного оборудования; стандарты информационного взаимодействия систем; методики тестирования разрабатываемого ПО; локальные нормативные правовые акты, действующие в		затруднения при оперировании знаниями при их переносе на новые ситуации.		

организации; государственные стандарты Единой системы программной документации (ЕСПД); способы адресации памяти целевой аппаратной платформы; технологии разработки компиляторов; методы и основные этапы трансляции; технологии программирования и разработки блок-схем; основы применения теории алгоритмов; принципы организации, состав и схемы работы операционных систем; основы архитектуры, устройства и функционирования вычислительных систем; средства программирования и их классификация; архитектура сред программирования; классификация языков программирования; основные структуры данных; основные модели данных и их организация; принципы объектно-ориентированного				

программирования; языки функционального и логического программирования; конкурентное программирование; методы и алгоритмы грамматического разбора текста программы; компиляторы языков программирования, их виды, принципы работы; методы и алгоритмы генерации исполняемого кода; структура объектных и исполняемых файлов в целевой операционной системе; компоновщики, методы сборки исполняемых файлов из объектных файлов; методы и алгоритмы оптимизации исполняемого кода; интерпретаторы языков программирования, их виды, принципы работы; методы управления памятью; принципы работы программного отладчика. ИПК-5.2. Умеет: применять языки программирования				

, определенные в техническом задании на разработку драйвера, для написания программного кода; создавать блок-схемы алгоритмов функционирования разрабатываемых программных продуктов; оценивать вычислительную сложность алгоритмов функционирования разрабатываемых программных продуктов; работать со стандартными контроллерами устройств (графическим адаптером, клавиатурой, мышью, сетевым адаптером); Работать с документацией, прилагаемой разработчиком устройства; осуществлять отладку драйверов устройств для операционной системы; применять языки целевой аппаратной платформы, определенной в техническом задании на разработку, для написания программного				

кода; применять технологию разработки компиляторов; осуществлять отладку программных продуктов для целевой операционной системы; осуществлять отладку утилит операционной системы; разрабатывать программный код на языках программирования низкого уровня; осуществлять отладку программ, написанных на языке программирования низкого уровня; разрабатывать программный код на языках программирования высокого уровня; осуществлять отладку программ, написанных на языке программирования высокого уровня; оформлять техническую документацию. ИПК-5.3. Владеет: получением и изучением технической документации устройства, для которого разрабатывается драйвер; разработкой блок-схем драйвера				

устройства, компиляторов, загрузчиков, сборщиков; написанием исходного кода драйвера устройства; отладкой разработанного драйвера устройства, компиляторов, загрузчиков, сборщиков; разработкой эксплуатационной документации на разработанных драйверов, компиляторов, загрузчиков, сборщиков; сопровождением разработанных драйвера устройства, компиляторов, загрузчиков, сборщиков; получением технической документации по языку программирования, системе команд процессора устройства, адресации памяти и регистров процессора устройства; изучением технической документации по языку программирования, системе команд процессора устройства, адресации памяти				

и регистров процессора устройства; разработка блок-схемы компиляторов, загрузчиков, сборщиков; написание исходного кода компиляторов, загрузчиков, сборщиков; отладка компиляторов, загрузчиков, сборщиков; реинжиниринг разработанных компиляторов, загрузчиков, сборщиков, драйвера устройства.				

Шкала оценивания результатов промежуточной аттестации и её описание:

Форма промежуточной аттестации: экзамен.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине.

Шкала оценивания	Описание
Отлично	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 5. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 4. Выполнены все виды

	учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 3. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не достигнуто пороговое значение хотя бы для одного уровня формируемых на момент проведения аттестации компетенций. Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7.3 Оценочные средства

7.3.1 Экзаменационные вопросы

1. Жизненный цикл ПО.
2. DevOps-инженер – роль в проекте разработки и внедрения ПО.
3. Экосистема контейнеров на основе Docker.
4. Автоматизация разработки, тестирования и доставки ПО с использованием Jenkins.
5. Основы управления конфигурацией с использованием Ansible.
6. Основные встроенные модули Ansible.
7. Взаимодействие Ansible с Docker.
8. Основы оркестрации с использованием Kubernets.
9. Микросервисная архитектура.
10. Взаимодействие Kubernets с Docker.
11. Особенности сбора метрик с микросервисов и Docker-контейнеров.